

JP9146751

Publication Title:

WINDOW DISPLAY METHOD AND WINDOW MANAGEMENT DEVICE

Abstract:

Abstract of JP 9146751

(A) Translate this text PROBLEM TO BE SOLVED: To provide a window display method and its managing device which make it possible to display information in a plurality of invariably visual window at the same time. SOLUTION: This device is equipped with a memory 1014 which stores attributes indicating the positions of respective windows in stack order, an input reception part 1020 which receive input indicating their changes, and a window display part 1018. The display part 1018 is equipped with an overlap processing part 1028 which determine whether or not a plurality of invariably visible windows are made to overlap with each other, an overlap alteration part 1030 which inhibits the invariably visual windows from overlapping with each other, a stack order processing part 1032 which determines whether or not a change gives a lower stack position to an invariably visible window, and a stack order change part 1034 which inhibits the invariably visible window from being given the lower stack position; when a plurality of windows share a common screen area, the part of a window which is stacked above is displayed in the common screen area.

Courtesy of <http://v3.espacenet.com>

【特許請求の範囲】

【請求項1】 情報を表示装置のウィンドウに表示するウィンドウ表示方法であって、

前記表示装置に表示された第1の常時可視ウィンドウを含む複数のウィンドウに情報を同時に表示するステップと、

前記複数のウィンドウの位置と形状を管理するステップと、

前記複数のウィンドウのうちの第2のウィンドウの部分と前記第1の常時可視ウィンドウの部分とに共通領域を生じせるイベントを検出するステップと、

前記第2ウィンドウが常時可視ウィンドウであるか否かを判定し、前記第2のウィンドウが常時可視ウィンドウである場合、第1の常時可視ウィンドウの又は／及び第2のウィンドウの形状を変更して、前記第2のウィンドウの部分と前記第1の常時可視ウィンドウの部分に重なる共通領域を生じさせないようにするステップとを有するウィンドウ表示方法。

【請求項2】 前記第2のウィンドウが常時可視ウィンドウでない場合、前記第1のウィンドウの前記部分を前記共通領域に表示するステップを有することを特徴とする請求項1に記載のウィンドウ表示方法。

【請求項3】 前記第1のウィンドウが透明背景を備え、

前記第1のウィンドウの前記部分を前記共通領域に表示するステップと、

前記第1のウィンドウとオーバーラップする不明確なウィンドウからの情報を前記共通領域に表示するステップとを有することを特徴とする請求項1に記載のウィンドウ表示方法。

【請求項4】 前記複数のウィンドウに関するスタック順序を管理するステップと、

ユーザ入力を受信して、前記第1の常時可視ウィンドウより上位のスタック位置を前記第2のウィンドウに与えるステップと、

前記第2のウィンドウが常時可視ウィンドウでない場合、前記スタック順序を変更して、前記第1の常時可視ウィンドウに、前記第2のウィンドウより上位のスタック位置を前記第1の常時可視ウィンドウに与えるステップとを有することを特徴とする請求項1に記載のウィンドウ表示方法。

【請求項5】 前記複数のウィンドウの各々がオーバーライド・アトリビュートを備え、

前記第2のウィンドウが常時可視ウィンドウであるか否か決定する前記ステップはオーバーライド・アトリビュートが設定されるか否か決定するステップを含んでいることを特徴とする請求項1に記載のウィンドウ表示方法。

【請求項6】 前記複数のウィンドウがルート・ウィンドウを含み、

前記ルート・ウィンドウが常時可視ウィンドウ・リストとして定められた特性を備え、

前記常時可視ウィンドウ・リストが常時可視ウィンドウである前記複数のウィンドウの各々に相応するエントリを備えており、

前記第2のウィンドウが常時可視ウィンドウであるか否か決定する前記ステップが、前記常時可視ウィンドウ・リストが前記第2のウィンドウに相応するエントリを含んでいるか否か決定するために前記常時可視ウィンドウ・リストを検査するステップを含んでいることを特徴とする請求項1に記載のウィンドウ表示方法。

【請求項7】 前記第1の常時可視ウィンドウを壊すイベントを検出し、前記常時可視ウィンドウ・リストから前記第1の常時可視ウィンドウに対応するエントリを除外するステップを備えることを特徴とする請求項6に記載のウィンドウ表示方法。

【請求項8】 新しい常時可視ウィンドウを作成するイベントを検出するステップと、

前記新しい常時可視ウィンドウに対応するエントリを前記常時可視ウィンドウ・リストに加えるステップと、

前記新しい常時可視ウィンドウが任意の他の常時可視ウィンドウとオーバーラップするか否かを決定するステップと、

前記新しい常時可視ウィンドウが任意の他の常時可視ウィンドウとオーバーラップする場合、新しい常時可視ウィンドウの又は／及び他の常時可視ウィンドウの形状を変更して、前記新しい常時可視ウィンドウの部分が前記他の常時可視ウィンドウとオーバーラップしないようにするステップとを備える特徴とする請求項6に記載のウィンドウ表示方法。

【請求項9】 ウィンドウ・ベース・インタフェースの情報をコンピュータ・システムの表示装置の画面に表示するウィンドウ表示方法であって、

少なくとも1の常時可視ウィンドウを含む複数のウィンドウを前記画面に表示するステップと、

重要な情報を前記複数のウィンドウの常時可視ウィンドウに表示するステップと、

前記複数のウィンドウの選択されたウィンドウに、前記常時可視ウィンドウの部分を重ねさせるユーザ入力を受信するステップと、

前記選択されたウィンドウが常時可視ウィンドウである場合、前記選択されたウィンドウを前記常時可視ウィンドウの重ねさせることを禁止するステップと、前記選択されたウィンドウが常時可視ウィンドウでない場合、前記選択されたウィンドウの最上部の前記常時可視ウィンドウを前記画面に表示するステップとを備えるウィンドウ表示方法。

【請求項10】 前記少なくとも1つの常時可視ウィンドウの各々に対応する画面の領域を指示する方向データを保存し、

前記選択されたウィンドウを前記画面の新しい領域に表示する試みが行われる時を検出し、新しい領域と方向データを比較して、新しい領域が任意の前記少なくとも1つの常時可視ウィンドウに対応する画面の領域と交差するか否かを決定することによって、前記選択されたウィンドウを任意の前記少なくとも1つの常時可視ウィンドウとオーバーラップさせる時を検出するステップを備えることを特徴とする請求項9に記載のウィンドウ表示方法。

【請求項11】 前記複数のウィンドウに関するスタック順序を維持するステップと、前記複数のウィンドウを前記スタック順序に相応して前記画面に表示し、前記複数のウィンドウの任意のウィンドウがオーバーラップする場合、上位のスタック位置を有するウィンドウをオーバーラップするウィンドウの上方に表示するステップと、前記ユーザ入力により常時可視ウィンドウでない任意のウィンドウに任意の前記少なくとも1つの常時可視ウィンドウより上位のスタック位置を与えることを検出するステップと、前記任意のウィンドウに前記少なくとも1つ常時可視ウィンドウの全てより下位のスタック位置を与えるステップとを備えることを特徴とする請求項9に記載のウィンドウ表示方法。

【請求項12】 前記常時可視ウィンドウのオーバーライド・アトリビュートを設定することによって、前記常時可視ウィンドウを常時可視ウィンドウとして指示するステップを備えることを特徴とする請求項9に記載のウィンドウ表示方法。

【請求項13】 常時可視ウィンドウ・リストを作成し、前記常時可視ウィンドウ・リストに前記常時可視ウィンドウに対応するエントリを加えることによって、前記常時可視ウィンドウを常時可視ウィンドウとして指示するステップを更に備えることを特徴とする請求項9に記載のウィンドウ表示方法。

【請求項14】 常時可視ウィンドウ・リストを作成する前記ステップが、前記常時可視ウィンドウ・リストをルート・ウィンドウの特性として構築するステップを含んでいることを特徴とする請求項13に記載のウィンドウ表示方法。

【請求項15】 前記エントリが、前記常時可視ウィンドウに関するウィンドウ識別子とロケーション指示要素とサイズ指示要素とスタック順序指示要素とを含んでいることを特徴とする請求項13に記載のウィンドウ表示方法。

【請求項16】 前記エントリが、前記常時可視ウィンドウがマッピングされるか否か指示するデータを含んでいることを特徴とする請求項15に記載のウィンドウ表示方法。

【請求項17】 表示装置を搭載するコンピュータ・シ

ステム上で使用するウィンドウ管理装置であって、複数のウィンドウの各々ウィンドウが常時可視ウィンドウであるか否かを示すと共に、前記複数のウィンドウの各々ウィンドウに関するスタック順序の位置を示す複数の属性を記憶する記憶装置と、

前記複数の属性の変化を指示する入力を受信する入力受信部と、

前記複数のウィンドウを前記表示装置に前記複数の属性に基づいて表示するウィンドウ表示部とを備え、

前記ウィンドウ表示部は、前記変化が前記2つ以上の常時可視ウィンドウをオーバーラップさせるか否かを決定するオーバーラップ処理部と、前記複数の属性を変更して前記2つ以上の常時可視ウィンドウ間のオーバーラップを禁止するオーバーラップ変更部と、前記変化が常時可視ウィンドウに常時可視ウィンドウでない任意のウィンドウより下位のスタック位置を与えるか否かを決定するスタック順序処理部と、前記複数の属性を変更して前記常時可視ウィンドウに常時可視ウィンドウでない任意のウィンドウより下位のスタック位置を与えることを禁止するスタック順序変更部とを備え、前記複数のウィンドウの2つ以上のウィンドウの部分が共通画面領域を共有する場合、前記共通画面領域に、前記2つ以上のウィンドウより前記上位にスタックされた位置を有する前記2つ以上のウィンドウのウィンドウの部分を表示することを特徴とするウィンドウ管理システム。

【請求項18】 どの前記複数のウィンドウが常時可視ウィンドウであるかについての記録を維持する手段を備えることを特徴とする請求項17に記載のウィンドウ管理システム。

【請求項19】 前記複数の属性がオーバーライド・アトリビュートを含んでおり、前記手段が、前記複数のウィンドウの各々のオーバーライド・アトリビュートを設定し、前記ウィンドウが常時可視ウィンドウであるか否かを反映させることにより、前記記録を維持することを特徴とする請求項18に記載のウィンドウ管理システム。

【請求項20】 前記複数のウィンドウがルート・ウィンドウを含んでおり、

前記ルート・ウィンドウが常時可視ウィンドウ・リストとして定められた特性を備え、

前記手段が、常時可視ウィンドウである前記複数のウィンドウの各々に相応する前記常時可視ウィンドウ・リストにエントリを保存することによって、前記記録を維持することを特徴とする請求項18に記載のウィンドウ管理システム。

【請求項21】 少なくとも1の常時可視ウィンドウを含む複数のウィンドウを前記画面に表示するステップと、重要な情報を前記複数のウィンドウの常時可視ウィンドウに表示するステップと、前記複数のウィンドウの選択されたウィンドウに、前記常時可視ウィンドウの部

分を重ねさせるユーザ入力を受信するステップと、前記選択されたウィンドウが常時可視ウィンドウである場合、前記選択されたウィンドウを前記常時可視ウィンドウの重ねさせることを禁止するステップと、前記選択されたウィンドウが常時可視ウィンドウでない場合、前記選択されたウィンドウの最上部の前記常時可視ウィンドウを前記画面に表示するステップをプロセスに実行させるシーケンスを記憶した記憶領域を有することを特徴とするコンピュータ読取可能情報記憶媒体。

【請求項22】 複数のウィンドウ・クラスからのウィンドウをサポートするウィンドウ・ベース・コンピュータ・システムであって、複数のウィンドウ・クラスの前記ウィンドウ・クラスが任意の他のウィンドウによって不明確にされないウィンドウに対応していることを特徴とするウィンドウ・ベース・コンピュータ・システム。

【請求項23】 前記特定のウィンドウ・クラスの任意の2つのウィンドウ間のオーバーラップを禁止するように構成したウィンドウ表示部を備えることを特徴とする請求項22に記載のウィンドウ・ベース・コンピュータ・システム。

【請求項24】 前記ウィンドウ表示部が前記特定のウィンドウ・クラスのウィンドウと前記特定のウィンドウ・クラスでないウィンドウとの間のオーバーラップを可能にするように構成されていることを特徴とする請求項23に記載のウィンドウ・ベース・コンピュータ・システム。

【請求項25】 前記ウィンドウ表示部が、前記特定のウィンドウ・クラスのウィンドウと前記特定のウィンドウ・クラスでないウィンドウとの間のオーバーラップを検出し、前記特定のウィンドウ・クラスでないウィンドウの最上部に前記特定のウィンドウ・クラスのウィンドウを表示するように構成されていることを特徴とする請求項24に記載のウィンドウ・ベース・コンピュータ・システム。

【請求項26】 前記ウィンドウ表示部は、前記2つのウィンドウをオーバーラップさせると思われる動作を行う時に、前記2つのウィンドウが互いに衝突するように前記ウィンドウを出現させることによって、前記特定のウィンドウ・クラスの2つのウィンドウ間のオーバーラップを禁止することを特徴とする請求項23に記載のウィンドウ・ベース・コンピュータ・システム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、コンピュータ・システムのウィンドウ管理、特に、Xウィンドウシステムの常時可視ウィンドウ階層を表示するウィンドウ表示方法及びウィンドウ管理装置に関する。

【0002】

【従来の技術】ウィンドウは、特定のデータ・セットと

特定の画面領域の相互関係を確定するユーザ・インタフェース・オブジェクトである。ウィンドウが別のユーザ・インタフェース・オブジェクト（ウィンドウ）に隠されたり、覆われなければ、データ・セットは、対応する画面領域に表示される。ウィンドウを用いることにより、コンピュータ・ユーザに情報を効果的に伝えることができることは明らかである。例えば、ユーザが複数の文書を開くことができるワード・プロセッシング装置では、ユーザは、複数の文書を開くことができ、開いた文書の各内容（コンテンツ）がそれぞれ別々のウィンドウに表示される。

【0003】多くのウィンドウ管理システムでは、柔軟性を高めるために、複数のウィンドウを互いにオーバーラップして表示することができるようになっている。それぞれのウィンドウが表示されるように設定された画面領域間に重なりがあるとき、2つのウィンドウは、オーバーラップして表示される。2以上のオーバーラップしたウィンドウの内容が、同一の画面領域に表示されると、情報（内容）が重なり、読みとことが困難になる。それゆえ、ウィンドウ管理システムは、オーバーラップするウィンドウに共通の画面領域に、どちらの内容を表示するかを決定しなければならない。

【0004】ウィンドウのオーバーラップの問題を解決するために、多くのウィンドウ管理システムは、各ウィンドウに“スタック順序”を設定するようになっている。複数のウィンドウが同じ画面領域を共有するとき、スタック順序の上位にスタックされたウィンドウの情報だけがその画面領域に表示される。したがって、共通の画面領域に表示されるはずの下位ウィンドウの部分は、共通の画面領域に対応する上位ウィンドウの部分によって“カバー”されることになる。ユーザにとっては、下位ウィンドウがあたかも上位ウィンドウの物理的下方に位置するように見える。

【0005】殆どのウィンドウ管理システムでは、ユーザが個々のウィンドウの位置や大きさ（サイズ）を変えたり、スタック順序を変更することができるようになっている。アップル・コンピュータ（Apple Computer, Inc.）が市販しているシステム7・オペレーティング・システムでは、ユーザがマウス又はトラックボール等の入力装置を用いてウィンドウの一部をクリックすると、ウィンドウがスタック順序において最上位のポジション（位置）になる。また、ウィンドウの底部の一端をドラッグすると、ウィンドウの大きさが変更される。さらに、ウィンドウのタイトル・バーを表示画面の別の位置にドラッグすると、ウィンドウが移動される。

【0006】あるアプリケーションにおいて、ある情報がユーザから常に見える状態になっていることが重要な場合がある。しかし、ウィンドウのオーバーラップが可能なシステムにおいて、ウィンドウ上の重要な情報がユーザから隠れてしまうことがある。例えば、ユーザが、

スタック順序においてあるウィンドウの上位に別のウィンドウをスタックする操作を行うと、その情報は隠される。このような状況は、ユーザの種々の種類の操作によって生じる。例えば、ユーザは、既に隠されているウィンドウのスタック順序を変更して上位とすることがある。また、ユーザは、スタック順序の上位に位置しているウィンドウを、あるウィンドウを隠す画面位置に移動することもある。

【0007】

【発明が解決しようとする課題】ウィンドウに表示された重要な情報がユーザに対して隠れることを禁止する種々の試みが行われてきた。例えば、ウィンドウ用のPCツール(PC Tools for Windows)では、あるウィンドウを常時上位(always on top)とすることをユーザが指定できる、ウィンドウをベースとしたグラフィック・デスクトップ・コンピュータを提供している。常時上位ウィンドウは、通常のウィンドウに割り当てられるスタック順序の最上位のポジションよりも上位のポジションに割り当てられる。そこで、任意のウィンドウが常時上位ウィンドウとオーバーラップするように操作されると、常時上位ウィンドウのオーバーラップ部分が共通の画面領域において他のウィンドウをカバーすることになる。

【0008】このアプローチ(試み)は、特に重要な情報を含んでいるウィンドウが1つだけである場合に効果的である。しかし、複数の常時上位ウィンドウが必要になる場合、問題が生じる。ウィンドウ用のPCツールでは、2つのウィンドウが常時上位ウィンドウとして指定されると、それらの常時上位ウィンドウ同士の関係は、通常のウィンドウ同士の関係と変わらない。すなわち、一方の常時上位ウィンドウが、他の常時上位ウィンドウよりもスタック順序の上位のポジションを有することになる。2つの常時上位ウィンドウがオーバーラップするときに、上位ウィンドウが、共通画面領域において下位ウィンドウをカバーすることになる。したがって、下位ウィンドウの重ねられた部分の情報は、ユーザから見ることができなくなる。したがって、複数の常時上位ウィンドウが指定された場合には、特に重要な情報が常に表示されることは保証されない。

【0009】この問題を回避する1つの解決策は、同時には1つだけしか常時上位ウィンドウを指定できないようにすることである。例えば、フランス特許第2,693,810号に、他のウィンドウによって隠されない1つの絶対明示(non-obscurable)ウィンドウを用いたウィンドウ管理システムが記載されている。ユーザが第2のウィンドウを絶対明示ウィンドウとして指定すると、第1の絶対明示ウィンドウの絶対明示ウィンドウ指定は解除される。このシステムは、複数の絶対明示ウィンドウをサポートしていないので、利便性に限界がある。

【0010】そこで、本発明は、上述の事情に鑑みてな

されたものであり、情報が複数の常時可視ウィンドウに同時に表示されることを可能にするウィンドウ表示方法及びウィンドウの管理装置を提供することを目的とする。

【0011】また、本発明は、現行のXウィンドウ規格に適合した状態でウィンドウの常時可視クラスを実現したウィンドウ表示方法及びウィンドウ管理装置を提供することを目的とする。

【0012】さらに、ユーザが、常時可視アトリビュート(属性)と透明背景アトリビュートを含んだウィンドウ・アトリビュートを組み合わせることができるウィンドウ表示方法及びウィンドウ管理装置を提供することを目的とする。

【0013】

【課題を解決するための手段】本発明は、情報を表示装置のウィンドウに表示するウィンドウ表示方法であって、前記表示装置に表示された第1の常時可視ウィンドウを含む複数のウィンドウに情報を同時に表示するステップと、前記複数のウィンドウの位置と形状を管理するステップと、前記複数のウィンドウのうちの第2のウィンドウの部分と前記第1の常時可視ウィンドウの部分とに共通領域を生じせるイベントを検出するステップと、前記第2ウィンドウが常時可視ウィンドウであるか否かを判定し、前記第2のウィンドウが常時可視ウィンドウである場合、第1の常時可視ウィンドウの又は/及び第2のウィンドウの形状を変更して、前記第2のウィンドウの部分で前記第1の常時可視ウィンドウの部分に重なる共通領域を生じさせないようにするステップとを有することを特徴とする。

【0014】また、本発明は、ウィンドウ・ベース・インタフェースの情報をコンピュータ・システムの表示装置の画面に表示するウィンドウ表示方法であって、少なくとも1の常時可視ウィンドウを含む複数のウィンドウを前記画面に表示するステップと、重要な情報を前記複数のウィンドウの常時可視ウィンドウに表示するステップと、前記複数のウィンドウの選択されたウィンドウに、前記常時可視ウィンドウの部分を重ねさせるユーザ入力を受信するステップと、前記選択されたウィンドウが常時可視ウィンドウである場合、前記選択されたウィンドウを前記常時可視ウィンドウの重ねさせることを禁止するステップと、前記選択されたウィンドウが常時可視ウィンドウでない場合、前記選択されたウィンドウの最上部の前記常時可視ウィンドウを前記画面に表示するステップとを備えることを特徴とする。

【0015】また、本発明は、表示装置を搭載するコンピュータ・システム上で使用するウィンドウ管理装置であって、複数のウィンドウの各々ウィンドウが常時可視ウィンドウであるか否かを示すと共に、前記複数のウィンドウの各々ウィンドウに関するスタック順序の位置を示す複数の属性を記憶する記憶装置と、前記複数の属性

の変化を指示する入力を受信する入力受信部と、前記複数のウィンドウを前記表示装置に前記複数の属性に基づいて表示するウィンドウ表示部とを備え、前記ウィンドウ表示部は、前記変化が前記2つ以上の常時可視ウィンドウをオーバーラップさせるか否かを決定するオーバーラップ処理部と、前記複数の属性を変更して前記2つ以上の常時可視ウィンドウ間のオーバーラップを禁止するオーバーラップ変更部と、前記変化が常時可視ウィンドウに常時可視ウィンドウでない任意のウィンドウより下位のスタック位置を与えるか否かを決定するスタック順序処理部と、前記複数の属性を変更して前記常時可視ウィンドウに常時可視ウィンドウでない任意のウィンドウより下位のスタック位置を与えることを禁止するスタック順序変更部とを備え、前記複数のウィンドウの2つ以上のウィンドウの部分共通画面領域を共有する場合、前記共通画面領域に、前記2つ以上のウィンドウより前記上位にスタックされた位置を有する前記2つ以上のウィンドウのウィンドウの部分を表示することを特徴とする。

【0016】また、本発明に係るコンピュータ読取可能情報記憶媒体は、少なくとも1の常時可視ウィンドウを含む複数のウィンドウを前記画面に表示するステップと、重要な情報を前記複数のウィンドウの常時可視ウィンドウに表示するステップと、前記複数のウィンドウの選択されたウィンドウに、前記常時可視ウィンドウの部分を重ねさせるユーザ入力を受信するステップと、前記選択されたウィンドウが常時可視ウィンドウである場合、前記選択されたウィンドウを前記常時可視ウィンドウの重ねさせることを禁止するステップと、前記選択されたウィンドウが常時可視ウィンドウでない場合、前記選択されたウィンドウの最上部の前記常時可視ウィンドウを前記画面に表示するステップをプロセッサに実行させるシーケンスを記憶した記憶領域を有することを特徴とする。

【0017】さらに、本発明は、複数のウィンドウ・クラスからのウィンドウをサポートするウィンドウ・ベース・コンピュータ・システムであって、複数のウィンドウ・クラスの特定のウィンドウ・クラスが任意の他のウィンドウによって不明確にされないウィンドウに対応していることを特徴とする。

【0018】

【発明の実施の形態】以下、本発明の好適な実施の形態について図面を参照しながら説明する。なお、以下の説明では、本発明を完全に理解できるように、多くの特別な実施形態の詳細が記載されている。しかし、当業者は、これらの詳細な記載がなくとも、本発明を理解することができる。

【0019】本発明に係るウィンドウ表示方法及びウィンドウ管理装置は、例えば図1に示すような構成のコンピュータ・システム100に適用される。このコンピュ

ータ・システム100は、情報を送受するためのバス101と、このバス101に接続された情報を処理するプロセッサ102とを備える。また、コンピュータ・システム100は、プロセッサ102によって実行される情報と命令を記憶するバス101に接続されたメイン・メモリ104を備える。このメインメモリ104は、例えばランダム・アクセス・メモリ(RAM)、DRAM等から構成され、プロセッサ102による命令の実行中に、一時的な変数又は他の中間情報を記憶する。

【0020】また、コンピュータ・システム100は、プロセッサ102の静的な情報と命令を記憶するバス101に接続されたリード・オンリ・メモリ(ROM)106と、情報と命令を記憶するバス101に接続された記憶装置107とを備える。この記憶装置107は、例えば磁気ディスク装置、光ディスク装置等から構成される。

【0021】さらに、コンピュータ・システム100は、バス101を介して接続された、例えば陰極線管(CRT)からなる表示装置121を備え、この表示装置121は、情報をコンピュータ・ユーザに表示する。また、文字や数字を入力するキーボード122が、バス101を介してプロセッサ102に接続されており、情報やコマンドの選択をプロセッサ102に伝える。別のタイプの入力装置として、方向情報とコマンド選択をプロセッサ102に通信し、かつ表示装置121上のカーソルの動きを制御する、例えばマウス、トラックボール、カーソル方向キー等のカーソル・コントロール123がある。このカーソル・コントロール123は、2つの軸、すなわち第1の軸(例えばx軸)と第2の軸(例えばy軸)において2つの自由度を一般的に備え、平面上の位置を指定できる。

【0022】また、マウス等に代えて、スタイラス(stylus)やペンのような他の入力装置も画面上の位置等を指示するインターフェイスとして使用できる。ペン等を用いてタッチスクリーンに表示されるオブジェクトに接触すると、そのオブジェクトは選択される。同様に、ライト・ペンと光感知式スクリーンも表示オブジェクトの選択に使用できる。このような装置は、マウス又はトラックボールを搭載するシステムにおけるポイントアンドクリック(point and click)の代わりに一つの動作で選択位置と選択を行うことになる。ペン等とタッチスクリーン、あるいは、ライトペンと光感知式スクリーンは、周知の技術である。このようなシステムには、122のようなキーボードがなく、情報はペンのような書き込み素子をインターフェイスとしてペン先を経由して提供され、書き込まれたテキストは光学式文字認識(OCR)技術を用いて解読される。

【0023】本発明は、複数の常時可視ウィンドウを表示可能なアプリケーション・プログラムを実行する方法、及びプラットフォームとしてのコンピュータ・シス

テムの使用に関する。以下、本発明の好適な実施の形態を、Xウィンドウ・システムの使用に即して説明する。

【0024】Xウィンドウ・システムは、MIT Project Athena と Digital Equipment Corporation が最初に開発したものであり、現在X Consortiumによって管理されている。Xウィンドウ・システムは、Xlib Reference Manual, volumes 1 and 2, O'Reilly & Associates, Inc (1995) に詳細に説明されている。

【0025】Xウィンドウ・システムにおいて、様々な機能は、クライアント・アプリケーションとサーバ・アプリケーションとの間で実行される。クライアントとサーバのアプリケーションは、単一のプロセッサ又は単一のワークステーション上で動作しているか、又はネットワーク上で接続している複数のワークステーション上の複数のプロセッサ上で動作している。

【0026】(クライアント・アプリケーション) クライアント・アプリケーションは要求をサーバ・アプリケーションに送る。要求は、指定動作の実行に関する要求、又は情報に関する要求を含んでいる。サーバ・アプリケーションは、指定動作を実行するか、又は要求情報を含んでいるクライアント・アプリケーションに回答を送って、要求に対応する。

【0027】(サーバ・アプリケーション) 要求に応じるほかに、サーバ・アプリケーションは、イベント・メッセージとエラー・メッセージとをクライアント・アプリケーションに送る。イベント・メッセージは、あるイベントが発生したことをクライアント・アプリケーションに知らせる。一般的に、クライアント・アプリケーションは、あるイベント・メッセージの受信時にある動作を行うように設計されている。エラー・メッセージは、クライアントからの前の要求が無効だったことをクライアントに知らせる。

【0028】サーバ・アプリケーションは複合リソースに関するアブストラクションを保持している。このようなりソースとして、例えば、ウィンドウ、ビクスマップ、カラーマップ、カーソル、フォント、グラフィック環境等がある。リソースのアブストラクションは、リソースのアトリビュートを含んでいる。例えば、ウィンドウのアブストラクションとして、サイズ、位置、境界幅、ウィンドウのスタック順序(ウィンドウの“コンフィグレーション”)に関する情報がある。

【0029】アブストラクションはサーバに保管されるので、リソースを使用しているクライアントが何人いるかに関わらず、アブストラクションは一つあれば足りる。クライアントはアブストラクションに対し直接アクセスできないので、リソースに関する情報を検索あるいは操作するには、サーバ・アプリケーションにその要求を送らなくてはならない。クライアントがリソースに関する要求を送ると、クライアントは、リソースに対応する整数のIDからリソースを識別する。この場合、整数

のIDで通信が行われるので、複雑なアブストラクション情報を送る場合に比べ、通信データ量を削減できる。

【0030】(ウィンドウ・マネージャ) ウィンドウ管理のために特別に設計されたウィンドウ・マネージャと呼ばれるクライアント・アプリケーションがある。ウィンドウ・マネージャは、画面スペースとカラーマップとを含む画面上の物理的なリソース間の表示形式を調整する働きをする。一般的に、ウィンドウ・マネージャは、サーバ・アプリケーション上にウィンドウ・ユーザー・インターフェイスを生成し、そのインターフェイスによって、ユーザーはウィンドウを移動し、ウィンドウのサイズを変更し、別のアプリケーションを実行することができる。

【0031】クライアント・アプリケーションがウィンドウの環境設定(configuration)の変更を要求する時に、クライアント・アプリケーションは要求をサーバに送る。しかし、ウィンドウの環境設定はウィンドウ・マネージャが制御しているので、サーバ・アプリケーションは要求に応じない。要求に直接応じない代わりに、サーバ・アプリケーションは、ウィンドウの環境の再設定を求めるイベント・メッセージをウィンドウ・マネージャに伝える。ウィンドウ・マネージャは、環境再設定の要求で指定された変更を実行、変更、又は拒絶すべきか、といった判断を行う。ウィンドウ・マネージャが環境設定の変更を実行すべきであると判断すると、ウィンドウ・マネージャは環境再設定要求をサーバに送り、サーバは再設定要求に応じる。ウィンドウ・マネージャが環境再設定要求の要求内容を変更すべきであると判断すると、ウィンドウ・マネージャは、変更した環境再設定要求をサーバ・アプリケーションに送る。サーバが環境再設定要求に応じる前に、承認のためにウィンドウ・マネージャに再設定要求を送るプロセスは、サブストラクチャー・リダイレクション(substructure redirection)と呼ばれる。

【0032】(ウィンドウ・アトリビュート) 上述のように、サーバ・アプリケーションは、ウィンドウの環境のほか、ウィンドウが画面上でどのように表示され、どのように振る舞うかといったリソースの情報をデータとして保持している。この情報はウィンドウ・アトリビュートと呼ばれる。ウィンドウ・アトリビュートのなかで、オーバーライド・アトリビュートと呼ばれるものがある。オーバーライド・アトリビュートの設定によって、ウィンドウ・マネージャによる干渉を受けることなく、ウィンドウを画面上にマッピングすることが可能となる。

【0033】(常時可視ウィンドウ) 本発明は、クライアント・アプリケーションが、常に見ることができるウィンドウを作成して表示することを可能にする。複数の常時可視ウィンドウが、ウィンドウの常時可視可能な性質を失うことなく同時に存在できる。常時可視ウィンド

うは、任意の数の通常のウィンドウとも同時に存在できる。この場合、通常のウィンドウとは、常時可視ウィンドウでない全てのウィンドウを意味する。常時可視ウィンドウの画面上の振る舞いについて、図2～図8を参照しながら説明する。

【0034】(ウィンドウの振る舞い) 本発明の好適な実施の形態において、ウィンドウは2つの単純な規則に従って振る舞う。第1の規則は、通常のウィンドウは、常時可視ウィンドウより上位にスタックされることが絶対にないという規則である。その結果、常時可視ウィンドウの一部が通常のウィンドウによってカバーされたり或いは不明確にされることがない。第2の規則は、常時可視ウィンドウ同士は、互いにオーバーラップする位置に設定されることを禁止し合うという規則である。その結果、常時可視ウィンドウの一部が任意の他の常時可視ウィンドウによってカバーされたり、或いは不明確にされることがない。

【0035】これらの規則に基づくウィンドウの振る舞いを図2～図9に示す。図2は、2つの常時可視ウィンドウ204と206と2つの通常のウィンドウ208と210とを画面202上に同時に表示する表示装置200を示す。ポインタ212も画面202上に図示してある。ポインタ212は、ロケーションを画面202上に示すためにユーザが用いる。ポインタ212は、マウス又はトラックボールのようなユーザ入力装置のユーザ操作に相応して、画面202上を移動する。

【0036】図2に示す画面202の状態では、ウィンドウ210は、ウィンドウ208の一部とオーバーラップしてウィンドウ208をカバーしている。ウィンドウ204と206は、ウィンドウ210の部分とオーバーラップしてカバーしている。ウィンドウ204又は206は任意の他のウィンドウによってカバーされていない。

【0037】前述のように、2つのウィンドウがオーバーラップする時に、どのウィンドウが他のウィンドウをカバーするか否かの決定は、ウィンドウのスタック順序におけるポジションから決まる。スタック順序で上位のウィンドウが、オーバーラップ領域に表示されて、スタック順序の下位のウィンドウをカバーすることになる。したがって、図2の画面202の状態は、ウィンドウ210がウィンドウ208よりスタック順序に関して現在上位であることと、ウィンドウ204と206がウィンドウ210よりスタック順序に関して現在上位であることを示している。

【0038】ウィンドウのオーバーラップを引き起こす操作は様々である。例えば、ウィンドウを他のウィンドウとオーバーラップする位置へ移動する操作、あるいはウィンドウのサイズを他のウィンドウとオーバーラップする領域まで拡大する操作などがある。ウィンドウに対する操作はスタック順序を変える。別のウィンドウでカ

バーされているウィンドウをカバーするウィンドウよりスタック順序に関して上位にするように、スタック順序が変更されると、上位にされたウィンドウのカバーされていた部分が画面上に表示され、下位にされたウィンドウの一部又は全部をカバーする。ウィンドウ環境を採用する標準的なシステムにおいて、ウィンドウを操作可能にするには、そのウィンドウを選択しなければならない。選択のプロセスは、例えば、ユーザ入力装置を動作してポインタ212の位置を選択すべきウィンドウの画面上に表示されている部分に設定して、マウス・ボタンのクリックのような所定のユーザ入力動作により行われる。ウィンドウ環境を採用する標準的なシステムでは、ウィンドウを選択すると、ウィンドウがスタック順序の最上位に自動的に設定される。しかし、以下図3～図8を参照しながら説明する通り、本発明は、この標準的なウィンドウ設定の状態を変更して、常時可視ウィンドウをサポートする。

【0039】図3は、ユーザがウィンドウ210を選択した後の画面202を示す。いまカバーされているウィンドウ210の境界が点線で図示してある。上述のように、ウィンドウ環境を採用する標準的なシステムでは、ウィンドウ210を選択する事により、ウィンドウ210は、スタック順序に関して最上位のポジションにスタックされる。したがって、ウィンドウ210の全体が画面上に表示され、オーバーラップする他のウィンドウの部分カバーする。しかし、本発明によれば、ウィンドウ210のような通常のウィンドウは、スタック順序に関し、常時可視ウィンドウより上位のポジションには絶対に移動しない。したがって、ウィンドウ210を選択しても、ウィンドウ210は、スタック順序に関して、常時可視ウィンドウであるウィンドウ204と206の上位のポジションには移動しないことになる。それゆえ、ウィンドウ210の選択しても、画面202は図2に示す状態と同様、ウィンドウ204と206の部分がウィンドウ210についてオーバーラップする部分をカバーすることになる。このように、1つ又は複数の常時可視ウィンドウとオーバーラップする通常のウィンドウを選択しても、通常のウィンドウは、常時可視ウィンドウよりスタック順序に関して上位のポジションには移動しないことになる。

【0040】図4は、ユーザがウィンドウ208を選択して拡大した状況を示している。ウィンドウ環境を採用する標準的なシステムでは、208を選択すると、ウィンドウ208がスタック順序に関して最上位のポジションに移動する。しかし、本発明のウィンドウ状態の規則に基づいて、ウィンドウ208は、他の通常のウィンドウに対するスタック順序に関しては、最上位の位置に置かれるが、常時可視ウィンドウに対するスタック順序に関しては、下位の位置を保つ。そこで、ウィンドウ208が図4に示すようにサイズ変更された後に、画面20

2は図5に示すようになる。ウィンドウ208は、ウィンドウ210よりスタック順序に関して上位に置かれるが、ウィンドウ204と206より依然として下位に置かれる。従って、ウィンドウ204と206は依然としてウィンドウ208をカバーしているが、ウィンドウ208は、ここでウィンドウ210をカバーすることになる。

【0041】図6は、ユーザがウィンドウ206を画面202上の別の位置に移動して、ウィンドウ206とウィンドウ204をオーバーラップさせようと試みている状況を示している。上述のように、本発明では、複数の常時可視ウィンドウのオーバーラップを禁止している。そこで、ウィンドウ206はユーザが指示した位置には移動せず、ユーザが指示した位置に最も近い位置であって、常時可視ウィンドウ204とオーバーラップしない位置まで戻される。このようなウィンドウの振る舞いによって、2つの常時可視ウィンドウは、オーバーラップする代わりに互いに衝突しているように見える。図7は、図6に示す動作後のウィンドウ206の位置を示す。特に、ウィンドウ206は、ウィンドウ204の近くに位置しているが、オーバーラップしていない。常時可視ウィンドウ同士に関しては、スタック順序に関するポジションは、意味をなさない。

【0042】図8は、常時可視ウィンドウのサイズを変更して、別の常時可視ウィンドウとオーバーラップさせる試みを示している。図8では、ウィンドウ206のサイズを変更して、ウィンドウ204とオーバーラップさせるような試みが行われている。図6及び図7に関してした上述の例のように、本発明においては、ウィンドウ206のサイズ変更動作は、別の常時可視ウィンドウとオーバーラップしない位置に限定して応じられる。この結果を、図9に例示する。

【0043】移動或いはサイズの変更を試みられた常時可視ウィンドウを、他の常時可視ウィンドウとオーバーラップしない位置まで戻す方式は、本発明の1つの例にすぎない。例えば、常時可視ウィンドウを別の常時可視ウィンドウとオーバーラップさせるような操作を行うと、操作された常時可視ウィンドウを操作前の状態まで戻すことも可能である。また、常時可視ウィンドウ間のオーバーラップは、一方の常時可視ウィンドウの動きが、他方の常時可視ウィンドウを別の位置に押し込むことによっても禁止できる。この方法に従えば、ユーザが図9のウィンドウ206を上方に移動しようと試みると、ウィンドウ204が、それに沿って上方に移動する。ウィンドウ204が画面202の上端に達すると、このウィンドウ204と206の上方向へのそれ以上の位置の変更は、不可能となる。このような常時可視ウィンドウ間のオーバーラップを禁止する押し込み方式は、他のオーバーラップ技術が要求する場合より複雑なウィンドウ管理ルーチンを必要とする。この複雑さは、例え

ば、ウィンドウAがウィンドウBを押し込み、ウィンドウBがウィンドウCとウィンドウDを押し込むといった、一方の常時可視ウィンドウの動きが任意の数の他方の常時可視ウィンドウの動きを要求することから生じる。

【0044】本発明は、上述の常時可視ウィンドウのオーバーラップを禁止するための規則は任意であり特定の方式に限定されるものではない。

【0045】(全体的なシステムの説明)図10は上述のウィンドウの振る舞いを実現するコンピュータ・システム1000の機能ブロック図である。コンピュータ・システム1000は表示装置1002と入力装置1004とウィンドウ管理システム1006とを備えている。ウィンドウ管理システム1006は、表示装置1002に表示するウィンドウを管理する。図示の例では、表示装置1002には1008と1010と1012の3つのウィンドウが表示されている。

【0046】ウィンドウ管理システム1006は、メモリ1014とウィンドウ表示部1018と入力受信部1020を備える。メモリ1014は、表示装置1002に表示されるウィンドウの複数のアトリビュートWND1(1022)とアトリビュートWND2(1024)とアトリビュートWND3(1026)とを備えている。メモリ1014に記憶されている、属性アトリビュートWND1(1022)とアトリビュートWND2(1024)とアトリビュートWND3(1026)は、各々ウィンドウについて、スタック順序に関するポジションと、画面上の位置とサイズ(垂直方向と水平方向のどちらも含む)、及びウィンドウが常時可視ウィンドウであるか否かについて書き込まれている。

【0047】常時可視(always-visible)ウィンドウ・アトリビュートは、Xウィンドウ仕様で定められ、サポートされる通常のウィンドウ・アトリビュートではない。そのため、特別に常時可視ウィンドウ・アトリビュートを定めたシステムは、Xウィンドウ・システムの仕様から逸脱し、他のシステムとの互換性を保てなくなる。他のシステムとの互換性といった観点からは、常時可視ウィンドウ・アトリビュートは、ウィンドウ・アトリビュートに存在しない方が望ましい。したがって、本発明において、Xウィンドウ・システムに事前に存在していたアトリビュート又はプロパティあるいはその両方である。ある実施の形態では、Xウィンドウ仕様で定めたオーバーライド(override-redirect)アトリビュートが常時可視(always-visible)のフラグとして用いられている。また、別の実施の形態では、ルート・ウィンドウに常時可視ウィンドウを識別するためのリストがプロパティとして設定されている。以下、これらの実施の形態について詳細に説明する。

【0048】図10は、オーバーライド・アトリビュ

トを常時可視ウィンドウ・フラグとして用いている実施の形態を示す。図示する例では、アトリビュートWND1(1022)は、ウィンドウ1008が常時可視ウィンドウでなく、ウィンドウ1008がスタック順序の第3位であることを示すデータを含んでいる。属性アトリビュートWND2(1024)は、ウィンドウ1010が常時可視ウィンドウでなく、ウィンドウ1010がスタック順序の第2位であることを示すデータを含んでいる。属性アトリビュートWND3(1026)は、ウィンドウ1012が常時可視ウィンドウであり、ウィンドウ1012がスタック順序の第1位であることを示すデータを含んでいる。

【0049】ウィンドウ表示部1018は、メモリ1014に記憶されているアトリビュートWND1(1022)とアトリビュートWND2(1024)とアトリビュートWND3(1026)に基づいて、表示装置1002にウィンドウ1008とウィンドウ1010とウィンドウ1012とを表示させる。表示装置1002の2つ以上のウィンドウの部分がオーバーラップしている場合、ウィンドウ表示部1018は、表示装置1002に、オーバーラップしている領域に関しては、上位にスタックされたウィンドウを優先して表示させる。例えば、ウィンドウ1008と1010がオーバーラップした場合、ウィンドウ1010はスタック順序に関してウィンドウ1008より上位なので、オーバーラップ領域には、1010が表示される。

【0050】入力受信部1020は、入力装置1004に接続して、ユーザから、例えばメモリ1014に記憶して、あるアトリビュートの変化を指示するような入力を受信する。ウィンドウ表示部1018は、オーバーラップ処理部1028とオーバーラップ変更部1030とスタック順序処理部1032とスタック順序変更部1034とを備えている。

【0051】オーバーラップ処理部1028は、アトリビュートの変更が2つ以上の常時可視ウィンドウをオーバーラップさせるか否かを判断する。オーバーラップ変更部1030は、アトリビュートWND1(1022)とアトリビュートWND2(1024)とアトリビュートWND3(1026)を必要に応じて変更し、常時可視ウィンドウ間のオーバーラップを禁止する。スタック順序処理部1032は、常時可視ウィンドウではないウィンドウが、スタック順序に関し、常時可視ウィンドウより上位に移動されたか否かを判断する。スタック順序変更部1034は、アトリビュートWND1(1022)とアトリビュートWND2(1024)とアトリビュートWND3(1026)を必要に応じて変えて、常時可視ウィンドウが常時可視ウィンドウでないウィンドウよりスタック順序に関して下位の位置を有することを禁止する。

【0052】前述のように、オーバーラップ処理部10

28とオーバーラップ変更部1030は共同して、常時可視ウィンドウが他の常時可視ウィンドウにカバーされないことを保証する。同様に、スタック順序処理部1032とスタック順序変更部1034は共同して、常時可視ウィンドウが常時可視ウィンドウでない任意のウィンドウによってカバーされないことを保証する。したがって、表示装置1002に表示される常時可視ウィンドウが他の任意のウィンドウによってカバーされることは、絶対にはない。

【0053】(クライアント・ベースの実施) 上述のように、Xウィンドウ・システムにおいて、ウィンドウ・マネージャのクライアントが、ユーザ・インタフェースの振る舞いを定める規則を管理する。ウィンドウの動き又は表示される外観に影響をおよぼす全ての要求は、サブストラクチャ・リディレクションを介してウィンドウ・マネージャに送られる。現在のウィンドウ・マネージャは、前述の常時可視ウィンドウの振る舞いをサポートしていない。そこで、本発明のある実施の形態においては、常時可視ウィンドウのオーバーライド・アトリビュートを設定して、常時可視ウィンドウのサブストラクチャ・リディレクションを無効にする。オーバーライド・アトリビュートは常時可視ウィンドウに対してだけ設定されるので、オーバーライド・アトリビュートの値も、フラグとして作用して、常時可視ウィンドウと通常のウィンドウとを区別する。

【0054】本発明を実施するクライアント・アプリケーションが常時可視ウィンドウを識別すると、前述のように、クライアントはそれ自体の常時可視ウィンドウのリストを維持する。リストは、ウィンドウIDとウィンドウの位置とウィンドウの形状を含めた、各々常時可視ウィンドウに関する情報を備えている。

【0055】クライアントは、常時可視ウィンドウの可視性に影響を与える可能性のある全てのグラフィックイベントを識別する。グラフィックイベントが常時可視ウィンドウの一部又は全部をカバーするように働く場合、クライアントは、イベントを変更して、上述のウィンドウの振る舞いに関する規則を実施する。特に、通常のウィンドウがスタック順序の最上位に置かれた場合、クライアントは、全ての常時可視ウィンドウを通常のウィンドウよりスタック順序に関して上位にさせる要求を直ちに送る。

【0056】クライアントは、常時可視ウィンドウが移動又はサイズ変更された場合、新しい位置とサイズについて、他の全ての常時可視ウィンドウとの関係を調べる。ウィンドウの移動又はサイズの変更によって、常時可視ウィンドウが1つ又は複数の他の常時可視ウィンドウとオーバーラップする場合、クライアントは、オーバーラップしない限界の位置まで常時可視ウィンドウの位置又はサイズを再設定する要求を送る。

【0057】(ウィンドウ・マネージャ・ベースの実

施) 上述の、ウィンドウ・マネージャを避けてクライアントベースで常時可視ウィンドウを管理する実施の形態に続き、以下、ウィンドウ・マネージャ・ベースの実施の形態を示す。

【0058】クライアント・ベースにおける実施の形態に類似して、ウィンドウ・マネージャは、常時可視ウィンドウの部分をカバー又は不明確にする可能性のあるイベントを常時可視ウィンドウのリストを保持している。この実施の形態では、常時可視ウィンドウの部分を不明確にするとされる動作を他のクライアントが行うことを禁止するために、オーバーライド・アトリビュート以外のフラグが、常時可視ウィンドウを識別するために用いられる。例えば、常時可視ウィンドウのリスト (always-visible window list) が、ルート・ウィンドウにプロパティとして組み込まれる。

【0059】図11に、常時可視ウィンドウ・リスト1102が常時可視ウィンドウを識別するために用いられている実施の形態を示す。常時可視ウィンドウ・リスト1102は各々存在する常時可視ウィンドウに対応するウィンドウ識別子を備えている。さらに、常時可視ウィンドウ・リスト1102は、表示装置1002の唯一の常時可視ウィンドウであるウィンドウ1012に対応するエントリ1104を含んでいる。ユーザがウィンドウのサイズを変更する操作、或いはウィンドウを選択し又は移動する操作を行っている間、ウィンドウ表示部1018は、常時可視ウィンドウ・リスト1102にアクセスして、どのウィンドウが常時可視ウィンドウであるかを調べる。ウィンドウ表示部1018が常時可視ウィンドウ・リストを用いて常時可視ウィンドウを識別すると、ウィンドウ表示部1018と通常のXシステム・ウィンドウ・マネージャによる常時可視ウィンドウの共同制御が可能になる。

【0060】(Xクライアント・アプリケーション) 図12は、本発明に基づくXクライアント・アプリケーションの動作を示すフローチャートである。Xクライアントは、1つ又は複数のウィンドウを常時可視 (always visible) であるとして指定できるように設計されている。上述のように、ウィンドウを常時可視ウィンドウとして指定する方法は幾通りもある。図13と図14はウィンドウを常時可視ウィンドウとして指定する二通りの方法を示している。図13に示す方法によれば、オーバーライド・アトリビュートを有する全てのウィンドウが常時可視ウィンドウになる。図14に示す方法によれば、全ての常時可視ウィンドウのウィンドウIDのリストがルート・ウィンドウにプロパティとして組み込まれる。この方法では、常時可視ウィンドウを作成する各アプリケーションが、ウィンドウ作成時に、そのウィンドウIDを常時可視ウィンドウ・リストに加えることができる。後述するバンプ・ウィンドウ・マネージャは、常時可視ウィンドウが消去される時に、その常時可視ウィ

ンドウのIDを常時可視ウィンドウ・リストから自動的に削除する。

【0061】図12において、Xクライアントは、S1202でXサーバに接続する。XクライアントとXサーバとの接続は、XクライアントがXサーバにXリブ・コール (Xlib call) を送ることによって行われる。Xリブ・コールは、Xクライアントと指定されたXサーバとの間の通信パスを構築する。

【0062】S1204で、Xクライアントはウィンドウを作成する。S1208で、Xクライアントは、S1204で作成したウィンドウを常時可視ウィンドウとして指定する。上述のように、ウィンドウは幾通りもある方法のうちの1つに従って常時可視ウィンドウの指定を行う。

【0063】図13に示す方法によれば、ウィンドウのオーバーライド・アトリビュートが、S1216でセットされ、ウィンドウは常時可視ウィンドウとなる。

【0064】図14に示す方法においては、ルート・ウィンドウは、常時可視ウィンドウ・リストと定義されるプロパティを備え、S1218で、常時可視ウィンドウ・リスト・プロパティが検索される。S1220で、新たに作成したウィンドウに対応するウィンドウIDが、常時可視ウィンドウ・リストにエントリとして追加され、その追加を反映した常時可視ウィンドウ・リスト・プロパティの内容の変更がS1222で行われる。、常時可視ウィンドウ・リストに対応するルート・ウィンドウ・リスト・プロパティが変更される。

【0065】図12において、ウィンドウは、S1210でマッピングされ、該当するディスプレイに表示される。S1212で、クライアントは全てのクライアント処理を実行する。このステップはクライアントの動作を包括的に表している。このステップの中で、Xクライアントにおいて、実際に実行されるステップは、Xクライアント自身、ユーザ、アプリケーションといった要素によって多種多様なものとなる。

【0066】Xクライアント動作中に、常時可視ウィンドウに影響する種々のイベントが発生することが想定される。例えば、他のウィンドウ (常時可視ウィンドウと通常のウィンドウの両方) が、作成され、マップされ、移動され、サイズ再設定され、消去される。Xクライアントが全ての処理を終えると、Xクライアントは、ウィンドウをS1214で消去する。

【0067】(バンプ・ウィンドウ・マネージャ) 本発明の実施の形態においては、バンプ・ウィンドウ・マネージャ (Bump Window Manager) が、上述のウィンドウの振る舞いを実現するために用いられる。バンプ・ウィンドウ・マネージャは、Xウィンドウ・マネージャの構成要素になるか、又はXウィンドウ・マネージャから独立して、Xウィンドウ・マネージャと共存する。バンプ・ウィンドウ・マネージャの働きは、常時可視ウィ

ドウとして指示されたウィンドウを識別して監視することである。スタック順序が変更されると、パンプ・ウィンドウ・マネージャは、全ての常時監視ウィンドウが全ての他のウィンドウより上位にスタックされていることを確かめる。常時監視ウィンドウが移動又はサイズ変更されると、パンプ・ウィンドウ・マネージャは、他の常時監視ウィンドウとのオーバーラップについてチェックする。このようなオーバーラップが見つかったと、パンプ・ウィンドウ・マネージャは、移動やサイズの変更が行われた常時監視ウィンドウを再度移動し、又はウィンドウ・サイズを再度変更し、他の常時監視ウィンドウとオーバーラップしないようにする。パンプ・ウィンドウ・マネージャの動作について、図15～図19を参照しながら詳細に説明する。

【0068】図15は、パンプ・ウィンドウ・マネージャの動作を示すフローチャートである。S1302で、パンプ・ウィンドウ・マネージャはXサーバに接続する。接続は、パンプ・ウィンドウ・マネージャと指定されたXサーバとの間にパスを設定することで行われる。

【0069】S1304で、パンプ・ウィンドウ・マネージャは常時監視ウィンドウ・リストを初期設定する。図16は、S1304においてパンプ・ウィンドウ・マネージャが行うステップを詳細に示している。S1314で、パンプ・ウィンドウ・マネージャはウィンドウのリストについてXサーバに問い合わせる。S1316で、パンプ・ウィンドウ・マネージャは、ウィンドウのリストの常時監視ウィンドウを識別して、常時監視ウィンドウのリストを作成する。Xサーバに認識されるウィンドウのうち、どのウィンドウが常時監視ウィンドウであるかを識別するために、パンプ・ウィンドウ・マネージャが用いる方法は、常時監視ウィンドウの指定に用いた方法に依存する。例えば、オーバーライド・アトリビュートをフラグとして用いて常時監視ウィンドウを指定する場合、パンプ・マネージャは、Xサーバで識別した全てのウィンドウのオーバーライド・アトリビュートを検査する。常時監視ウィンドウ・リストをプロパティとしてルート・ウィンドウに組み込む場合、パンプ・ウィンドウ・マネージャは、ルート・ウィンドウの該当するプロパティを検査する。

【0070】S1318で、パンプ・ウィンドウ・マネージャは全ての常時監視ウィンドウのサイズと位置についてXサーバに問い合わせる。S1320で、パンプ・ウィンドウ・マネージャは、常時監視ウィンドウ・リストに、常時監視ウィンドウの各々のサイズと位置とステータス（マッピングされているか否か）を保存する。そこで、常時監視ウィンドウ・リストの各々常時監視ウィンドウのエントリは、常時監視ウィンドウのウィンドウIDと位置とサイズとステータスとスタック順序を備えていることになる。

【0071】S1322で、パンプ・ウィンドウ・マネ

ージャは新しいリストを処理する。新しい常時監視ウィンドウ・リストを処理するためにパンプ・ウィンドウ・マネージャが行う動作が図17に図示してある。図17を見ると、S1326とS1338とS1340は、マッピングした常時監視ウィンドウ間にオーバーラップが現れるまで、又は常時監視ウィンドウの全てのエントリが処理されるまでループする。

【0072】マッピングしてないウィンドウは、情報が表示されないで、別のウィンドウに隠されるといった問題は起こり得ない。従って、好適な実施の形態においては、マッピングしてない常時監視ウィンドウと任意の他の常時監視ウィンドウとの間のオーバーラップは、無視される。

【0073】オーバーラップが、マッピングされた常時監視ウィンドウと1つ又は複数の他のマッピングされた常時監視ウィンドウとの間に発生すると、ステップは、S1326からS1328に進む。S1328で、パンプ・ウィンドウ・マネージャは、常時監視ウィンドウが他の全てのマッピングされた常時監視ウィンドウとの間にオーバーラップしない位置を検索する。S1330で、このような位置が存在するか否かについて判断される。このような位置が存在すると、常時監視ウィンドウは、S1336でその位置に移動する。このような位置が存在しない場合、コントロールはS1332に進む。

【0074】S1332で、パンプ・ウィンドウ・マネージャは、常時監視ウィンドウのサイズ限界に達しているか否かを判断する。この場合、サイズ限界は、常時監視ウィンドウにパラメータとして設定されている。常時監視ウィンドウがサイズ限界に達していれば、ステップはS1338に進んで、リストの次の常時監視ウィンドウ・エントリが処理される。常時監視ウィンドウのサイズ限界に達していなければ、常時監視ウィンドウのサイズが縮小される。S1326、S1328、S1330、S1332とS1334は、ウィンドウがオーバーラップしなくなるまで、ループし、常時監視ウィンドウの移動又はそのサイズ限界までの縮小を繰り返す。

【0075】図15のS1306において、パンプ・ウィンドウ・マネージャは、常時監視ウィンドウ・リストの初期設定中、ウィンドウ・イベントを待機している。ウィンドウ・イベントが発生すると、コントロールは、ウィンドウ・イベントが実行されるS1308に進む。以下、S1308について、図18と図19を参照しながら詳細に説明する。

【0076】図18と図19は、ウィンドウイベントが発生した時に、パンプ・ウィンドウ・マネージャが行うステップのフローチャートを示している。S1342で、パンプ・ウィンドウ・マネージャは、発生したイベントを識別するデータを読みとる。S1344～S1358で、パンプ・ウィンドウ・マネージャは、発生したイベントのタイプを識別する。

【0077】イベントが特定のウィンドウが移動されることを示す場合、コントロールは、S1344からS1360に進む。S1360で、パンパ・ウィンドウ・マネージャは、動作が行われるウィンドウが常時可視ウィンドウであるか否かを判断する。ウィンドウが常時可視ウィンドウでない場合、パンパ・ウィンドウ・マネージャは、特別な処理を行わない。ウィンドウが常時可視ウィンドウである場合、コントロールはS1376に進む。S1376で、パンパ・ウィンドウ・マネージャは、その移動で常時可視ウィンドウが別の常時可視ウィンドウとオーバーラップするか否かを判断する。判断の結果、移動した常時可視ウィンドウが他の常時可視ウィンドウとオーバーラップしない場合、パンパ・ウィンドウ・マネージャは、特別な処理を行わない。S1376における判断の結果、移動した常時可視ウィンドウが他の常時可視ウィンドウとオーバーラップする場合、ステップはS1384に移り、常時可視ウィンドウは、オーバーラップしない位置に移動される。オーバーラップしない位置とは、例えば、他の常時可視ウィンドウに隣接しているがオーバーラップしない位置等である。S1390で、常時可視ウィンドウ・リストは更新され、移動したウィンドウの新しい位置のデータが追加される。

【0078】また、イベントが特定のウィンドウのサイズを変更するものである場合、コントロールはS1346からS1362に進む。S1362で、パンパ・ウィンドウ・マネージャは、動作が行われるウィンドウが常時可視ウィンドウであるか否かを決定する。ウィンドウが常時可視ウィンドウでない場合、パンパ・ウィンドウ・マネージャは、特別な処理を行わない。ウィンドウが常時可視ウィンドウである場合、コントロールはS1378に進む。S1378で、パンパ・ウィンドウ・マネージャは、常時可視ウィンドウのサイズの変更によって、その常時可視ウィンドウが別の常時可視ウィンドウとオーバーラップするか否かを判断する。S1378における判断の結果、サイズを変更された常時可視ウィンドウが他の常時可視ウィンドウとオーバーラップしない場合、パンパ・ウィンドウ・マネージャは、特別な処理を行わない。S1378における判断の結果、サイズを変更された常時可視ウィンドウが他の常時可視ウィンドウとオーバーラップする場合、ステップはS1386に移り、その常時可視ウィンドウを他の常時可視ウィンドウとオーバーラップしないサイズに変更する。当然、S1386で変更されたサイズは、イベントで指定されたサイズより小さくなる。サイズは、例えば、サイズ変更したウィンドウの常時可視ウィンドウの境界に隣接するがオーバーラップしないサイズに変更される。S1392で、常時可視ウィンドウ・リストは更新され、変更したウィンドウの新しいサイズのデータが追加される。

【0079】イベントがスタック順序を変更するもので

ある場合、ステップはS1348からS1364に進む。S1364で、パンパ・ウィンドウ・マネージャは、イベント発生後の新しいスタック順序において、常時可視ではないウィンドウが任意の常時可視ウィンドウの上位になっているか否かを判断する。常時可視ではないウィンドウが、スタック順序において常時可視ウィンドウの上位にない場合、パンパ・ウィンドウ・マネージャは、特別な処理を行わない。常時可視ではないウィンドウがイベント発生後の新しいスタック順序において常時可視ウィンドウの上位にある場合、ステップはS1380に進む。S1380で、パンパ・ウィンドウ・マネージャは、全ての常時可視ウィンドウがスタック順序に関して全ての常時可視ではないウィンドウの上位になるようにスタック順序を変更する。

【0080】イベントが特定のウィンドウを作成するものである場合、ステップはS1350からS1366に進む。S1366で、パンパ・ウィンドウ・マネージャは、新たに作成したウィンドウが常時可視ウィンドウであるか否かを判断する。ウィンドウが常時可視ウィンドウでない場合、パンパ・ウィンドウ・マネージャはS1382で、全ての常時可視ウィンドウが、スタック順序に関して、全ての常時可視ではないウィンドウの上位となるように、スタック順序を変更する。ウィンドウが常時可視ウィンドウである場合、ステップはS1388に進み、新たに作成した常時可視ウィンドウのエントリが常時可視ウィンドウ・リストに追加され、常時可視ウィンドウ・リストが更新される。

【0081】イベントが特定のウィンドウを消去するものである場合、ステップはS1352からS1368に進む。S1368で、パンパ・ウィンドウ・マネージャは、動作が行われるウィンドウが常時可視ウィンドウであるか否かを判断する。消去されたウィンドウが常時可視ウィンドウでない場合、パンパ・ウィンドウ・マネージャは、特別な処理を行わない。消去されたウィンドウが常時可視ウィンドウである場合、コントロールはS1396に進み、S1396で、常時可視ウィンドウ・リストが更新され、消去される常時可視ウィンドウのエントリが、常時可視ウィンドウ・リストから削除される。

【0082】イベントが特定のウィンドウをマッピングするものである場合、ステップはS1354からS1370に進む。S1370で、パンパ・ウィンドウ・マネージャは、マッピングされるウィンドウが常時可視ウィンドウであるか否かを判断する。マッピングされるウィンドウが常時可視ウィンドウでない場合、パンパ・ウィンドウ・マネージャは、特別な処理を行わない。マッピングされるウィンドウが常時可視ウィンドウである場合、ステップはS1398に移る。S1398では、イベントに関する常時可視ウィンドウのマッピングを反映する、常時可視ウィンドウ・リストの更新がなされ、常

常時可視ウィンドウのエントリにおけるステイタス（ウィンドウがマッピングされているか否かに関する）に変更が加えられる。エントリが変更されると、ステップはS1394に移り、常時可視ウィンドウ・リストは、マッピングした常時可視ウィンドウと、他のマッピングされている常時可視ウィンドウとのオーバーラップを防ぐ処理がなされる。この常時可視ウィンドウ・リストの処理の内容は図17を参照してすでに述べたものと同様である。

【0083】イベントが特定のウィンドウをマップ除去（すでにマッピングされているウィンドウをマッピングされていない状態にすること）するものである場合、ステップはS1356からS1372に進む。S1372で、パンプ・ウィンドウ・マネージャは、マップ除去が行われるウィンドウが常時可視ウィンドウであるか否かを判断する。マップ除去されるウィンドウが常時可視ウィンドウでない場合、パンプ・ウィンドウ・マネージャは、特別な処理を行わない。ウィンドウが常時可視ウィンドウである場合、ステップはS1400に進む。S1400では、イベントに関する常時可視ウィンドウのマップ除去を反映する。常時可視ウィンドウ・リストの更新がなされ、常時可視ウィンドウのエントリにおけるステイタス（ウィンドウがマッピングされているか否かに関する）に変更が加えられる。

【0084】イベントがウィンドウのプロパティを変更するものである場合、コントロールはS1358からS1374に進む。S1374で、パンプ・ウィンドウ・マネージャは、変更されるプロパティが常時可視ウィンドウ・リスト・プロパティであるか否かを判断する。変更される特性が常時可視ウィンドウ・リストでない場合、パンプ・ウィンドウ・マネージャは、特別な処理を行わない。変更されるプロパティが常時可視ウィンドウ・リストである場合、ステップはS1402に進む。S1402では、常時可視ウィンドウ・リストの内容が、イベントの指示に基づいて変更される。常時可視ウィンドウ・リストの内容が変更されると、常時可視ウィンドウ・リストは、S1404において、マッピングされている常時可視ウィンドウと、他のマッピングされている常時可視ウィンドウとのオーバーラップを防ぐ処理がなされる。この常時可視ウィンドウ・リストの処理の内容は図17を参照してすでに述べたものと同様である。

【0085】図15において、パンプ・ウィンドウ・マネージャが終了可能になるまで、S1306、S1308及びS1310ではループし、上述の処理を繰り返す。パンプ・ウィンドウ・マネージャが終了可能になると、ステップはS1310からS1312に進み、S1312で、パンプ・ウィンドウ・マネージャは、パンプ・ウィンドウ・マネージャとXサーバとの接続を切断する。

【0086】（背景透明性）本発明に背景透明性（Trans-

parent Background）を有する常時可視ウィンドウを用いることによって、さらに好適な実施の形態を得ることができる。背景透明性とは、ウィンドウに表示されるデータ以外の背景を透明にする性質である。背景透明性を有するウィンドウの背景にあたる画面領域では、カバーされているウィンドウの一部が表示される。背景ではない部分、つまりデータの部分は、透明性を有さず、画面に表示される。常時可視ウィンドウがその背景透明性を有することによって、特定のデータを常に画面に表示することを保証しながら、常時可視ウィンドウによって隠される情報の量を最小限にすることができる。

【0087】（可能性のあるアプリケーション）本発明が提供する常時可視ウィンドウは、重要な情報を常時表示しなくてはならないコンピュータ・アプリケーションにとって理想的である。例えば、航空交通管制において、航空機に関連するデータの中で、他の情報によってカバーされることが許されないデータがある。今日まで、この制約のために、航空交通管制アプリケーションはウィンドウ・ベース環境を活用できなかった。しかし、重要なデータを常時可視ウィンドウで表示することによって、航空交通管制アプリケーションも、ウィンドウ・ベースのグラフィック環境を活用できることとなる。航空交通管制は、本発明から大きな利益を得ることができる分野の1例にすぎない。本発明は、任意の特殊な使用分野に限定されるものではない。

【0088】以上、本発明の特別な実施の形態について説明してきたが、本発明は、上記実施の形態のみに限定されるものでなく、種々の変更と変形を含むものである。

【0089】

【発明の効果】以上のように、本発明によれば、コンピュータのウィンドウ環境にウィンドウの常時可視階層を構築可能なウィンドウ表示方法及びウィンドウ管理装置を提供することができる。これにより、ユーザは、1つ又は複数のウィンドウを常時可視ウィンドウとして指定できる。常時可視ウィンドウが常時可視ではないウィンドウとオーバーラップすると、常時可視ウィンドウは常時可視ではないウィンドウより上に表示される。また、常時可視ウィンドウは、互いにオーバーラップしないように構成される。

【0090】本発明によれば、Xウィンドウ・システムに適應する状態で常時可視階層の構築することができる。ある方式では、オーバーライド・アトリビュートが、どのウィンドウが常時可視ウィンドウあるかを識別するフラグとして用いられる。また、別の方式では、常時可視ウィンドウのリストがルート・ウィンドウにプロパティとして組み込まれる。

【0091】本発明によれば、情報を表示装置上の複数のウィンドウに同時に表示できる。複数のウィンドウが第1の常時可視ウィンドウを含んでいる。複数のウィン

ドウに対応する複数のウィンドウが維持される。特に、各々ウィンドウに相応する構成が、表示装置のウィンドウのロケーションと形状を反映するように維持される。

【0092】第2のウィンドウの部分に第1の常時可視ウィンドウの部分具备する表示装置の共通領域を占めさせるイベントが、検出される。このようなイベントを検出すると、第2のウィンドウも、常時可視ウィンドウであるか否かが決定される。第2のウィンドウが常時可視ウィンドウである場合、1つ又は両方のウィンドウの、ウィンドウ間のオーバーラップを禁止するために変更される。第2のウィンドウが常時可視ウィンドウでない場合、第1のウィンドウが第2のウィンドウに表示される。

【0093】本発明によれば、スタック順序が複数のウィンドウに相応して維持される。第2のウィンドウに、第1の常時可視ウィンドウより上位にスタックされた位置を持たせる、ユーザ入力が受信される。第2のウィンドウが常時可視ウィンドウでない場合、スタック順序は、第1の常時可視ウィンドウに第2のウィンドウより上位にスタックされた位置を持たせるために変更される。

【0094】本発明によれば、種々の技術がウィンドウを常時可視ウィンドウとして指示するために使用できる。ある実施の形態では、Xウィンドウ・システムに設けてあるオーバーラップ再検討属性が、常時可視ウィンドウを指示するフラグとして用いられる。別の実施の形態では、ルート・ウィンドウに組み込まれている特性が、常時可視ウィンドウ・リストとして用いられる。ウィンドウを常時可視ウィンドウとして指示するために、ウィンドウを識別するエントリが常時可視ウィンドウ・リストに加えられる。これらの技術は、共に、Xウィンドウ基準から逸脱せずに使用できる。

【図面の簡単な説明】

【図1】本発明が適用されるコンピュータ・システムの構成を示す図である。

【図2】画面上に2つの常時可視ウィンドウと2つの通常のウィンドウを同時に表示する表示装置の画面を図である。

【図3】ユーザが常時可視ウィンドウとオーバーラップする通常のウィンドウを選択した後の表示装置の画面を示す図である。

【図4】ユーザが通常のウィンドウを選択してサイズを拡大し通常のウィンドウを 常時可視ウィンドウにオーバーラップさせるプロセスにある時の、表示装置の画面を示す図である。

【図5】図4に示したサイズ変更動作が終了した時の表示装置の画面を示す図である。

【図6】ユーザが、常時可視ウィンドウを別の常時可視ウィンドウとオーバーラップする位置に移動しようとした時の表示装置の画面を示す図である。

【図7】常時可視ウィンドウが移動される様子を示す図6に示した動作が、別の常時可視ウィンドウとオーバーラップすることを許可されない後の、表示装置の画面を示す図である。

【図8】ユーザが、常時可視ウィンドウを別の常時可視ウィンドウとオーバーラップさせるように常時可視ウィンドウのサイズの変更を試みる時の、表示装置の画面を示す図である。

【図9】図8に示した動作の後の表示装置の画面を示す図である。

【図10】本発明に従う常時可視ウィンドウを実現するウィンドウ・コンピュータ・システムの機能構成を示すブロック図である。

【図11】本発明に従う常時可視ウィンドウを実現するウィンドウ・コンピュータ・システムの他の機能構成を示すブロック図である。

【図12】本発明に従うXクライアントの動作を示すフローチャートである。

【図13】本発明に従う常時可視ウィンドウを指定するステップを示すフローチャートである。

【図14】本発明に従う常時可視ウィンドウを指示するステップを示すフローチャートである。

【図15】本発明に従うパンプ・ウィンドウ・マネージャの動作を示すフローチャートである。

【図16】常時可視ウィンドウ・リストを初期化するために、パンプ・ウィンドウ・マネージャによって行われるステップを示すフローチャートである。

【図17】常時可視ウィンドウを処理して、常時可視ウィンドウ間のオーバーラップを防止するために、パンプ・ウィンドウ・マネージャによって行われるステップを示すフローチャートである。

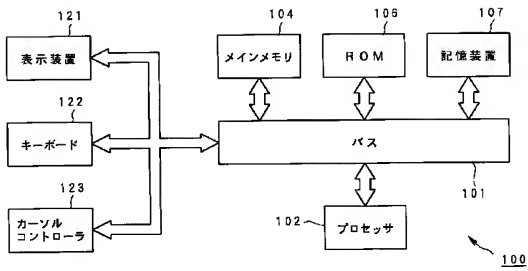
【図18】ウィンドウに影響するイベントの検出に相応して、パンプ・ウィンドウ・マネージャによって行われるステップを示すフローチャートである。

【図19】図18に示したフローチャートの続きを示すフローチャートである。

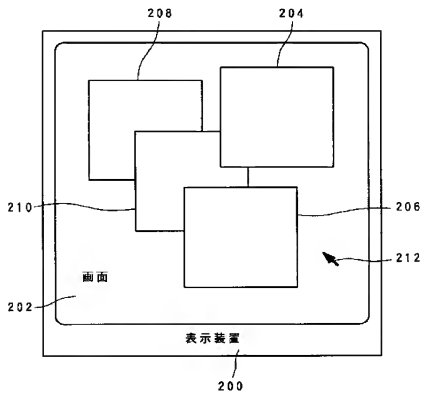
【符号の説明】

101 バス、102 プロセッサ、104、メインメモリ、107 記憶装置、121 ディスプレイ、122 キーボード、123 カーソル・コントローラ、200 表示装置、202 画面、1004 入力装置、1014 メモリ、1018 ウィンドウ表示部、1020 入力受信部、1022 アトリビュートWND1、1024 アトリビュートWND2、1026 アトリビュートWND3、1028 オーバーラップ処理部、1030 オーバーラップ変更部、1032 スタック順序処理部、1034 スタック順序変更部、1102 常時可視ウィンドウ・リスト、1104 ウィンドウ1012のエントリ

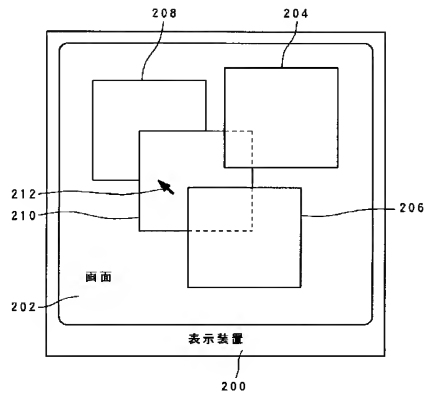
【図1】



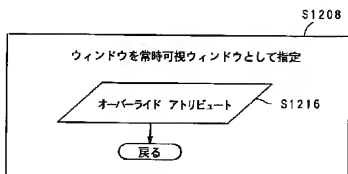
【図2】



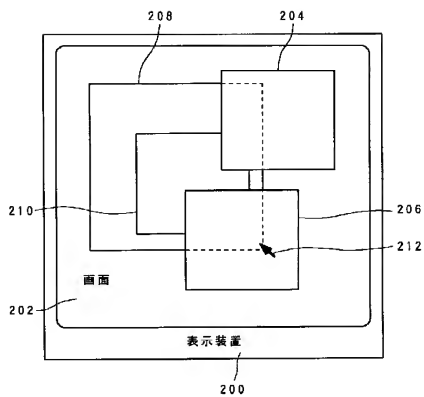
【図3】



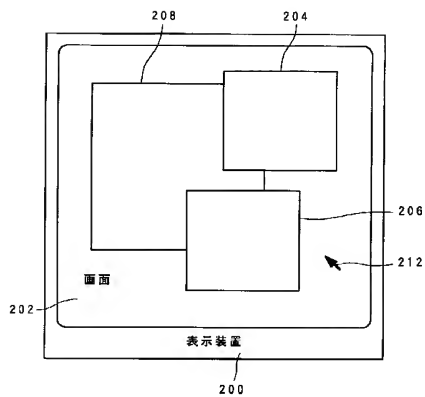
【図13】



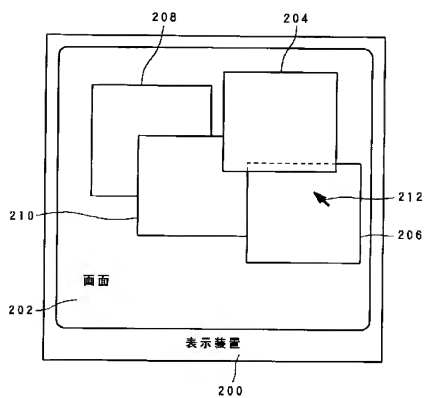
【図4】



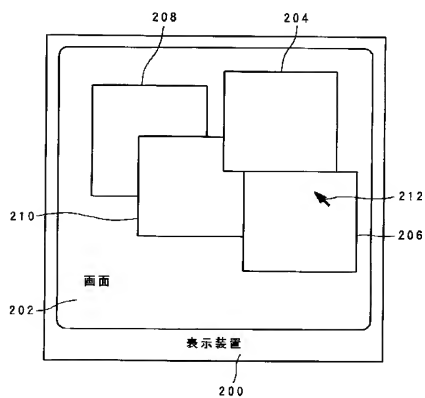
【図5】



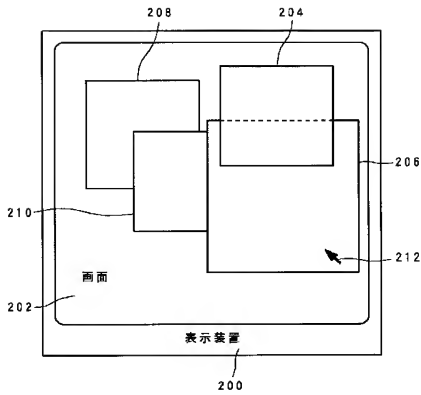
【図6】



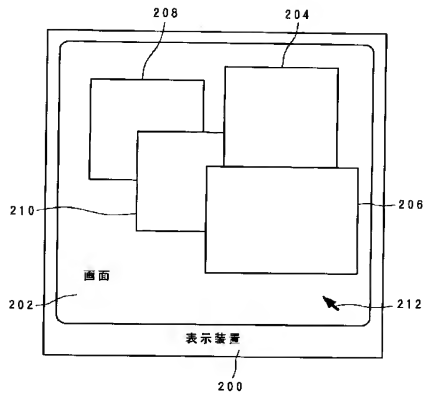
【図7】



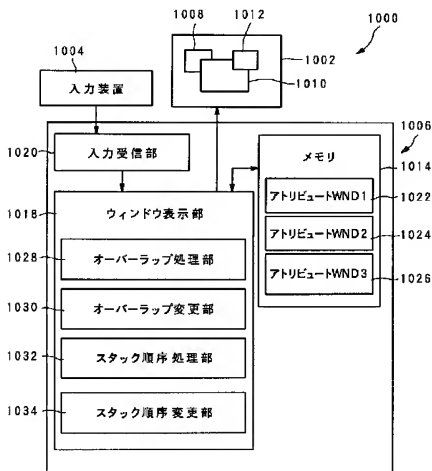
【図8】



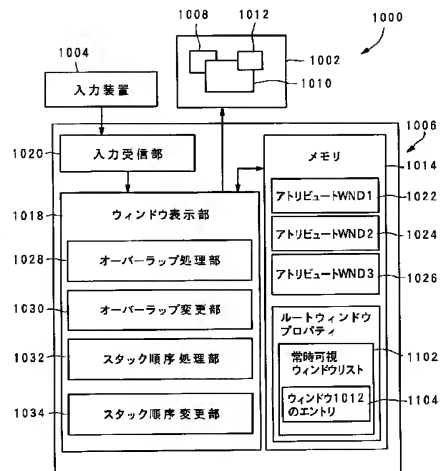
【図9】



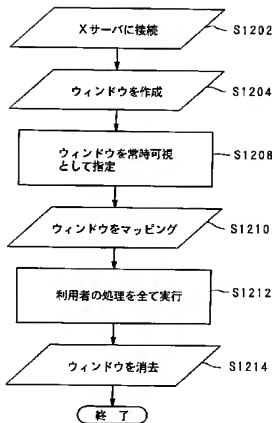
【図10】



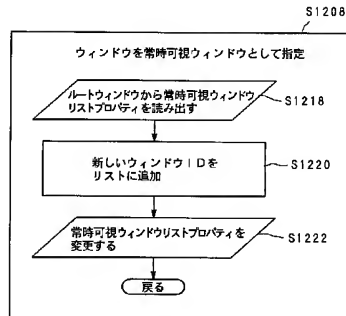
【図11】



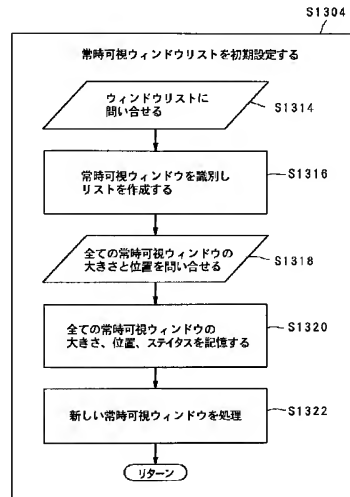
【図12】



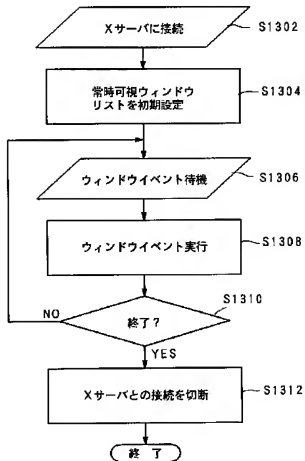
【図14】



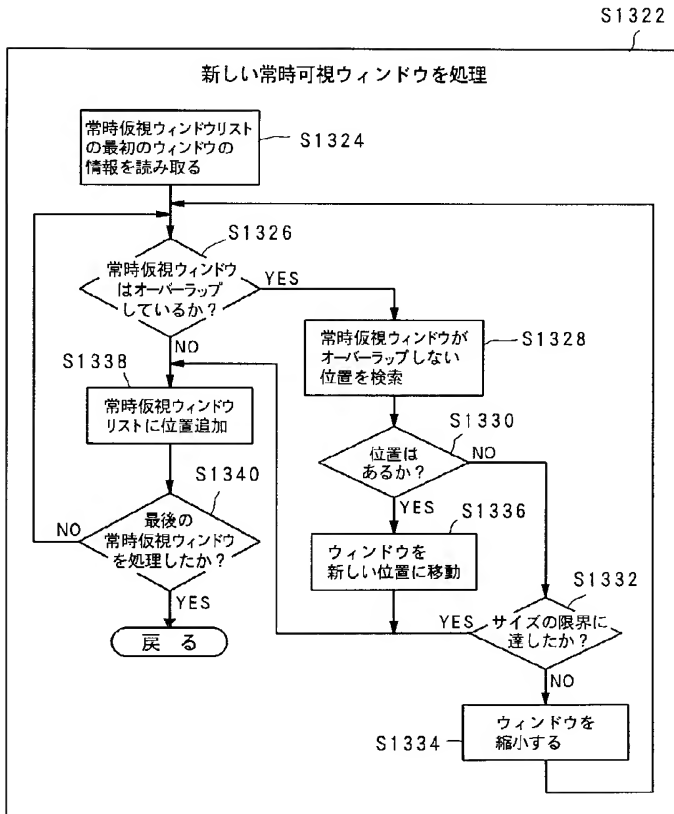
【図16】



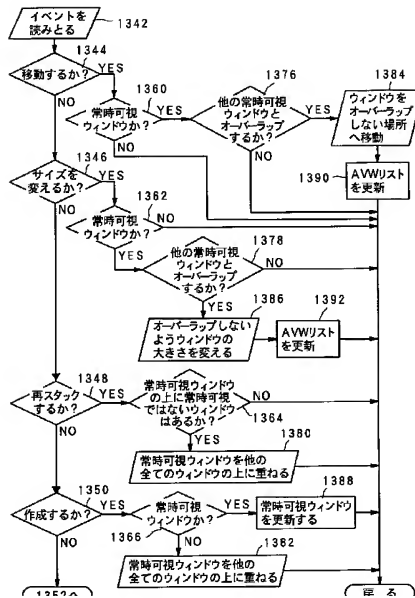
【図15】



【図17】



【図18】



【図19】

